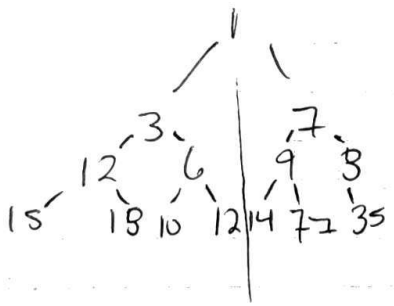
Yousef Awad

# Heap

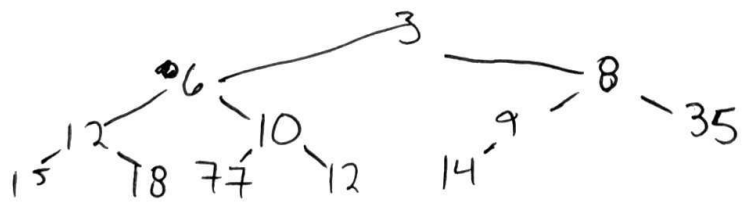
1) $2 \cdot (2i+1) + 1 = 4i + 2 + 1 = \boxed{4i+3}$

2) $2 \cdot (2i+2) + 2 = 4i + 4 + 2 = \boxed{4i+6}$

3)



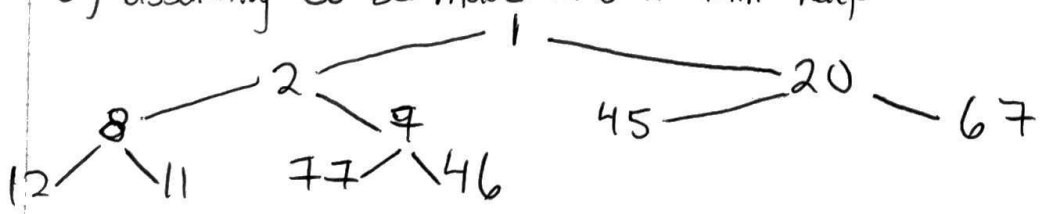

4)




5)

| 1 | 3 | 8 | 12 | 6 | 9 | 35 | 15 | 18 | 10 | 12 | 14 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

6) assuming to be made into a min heap:



#1

```
void percolateDown(struct heapStruct *h, int index){
  int mini ← minimum index of minimum child by value
  if((2*index+1) <= h->size) ← let the index we are percolating has both children
      min = minimum(h->heap....); ← finding the minimum value's index between the given index
      if(h->heaparray[index] > h->heap[min]) ← if the current index's value is larger than the more,
          swap(index, min); ← swap them
          percolateDown(min); ← call recursively on min index.
  }else if(h->size == 2*index) { ← if the index has only one child.
      if(h->heap[index] > h->heap[2*index]){ ← and if the current index's value > its child's
          swap(index, 2*index); ← swap them
      }
}
```